РОЗДІЛ 2

РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ РОБОТИ ФЕС

2.1. Загальний огляд компонентів системи

Так як метою даного дипломного проекту є створення власного аналога моніторингової системи роботи ФЕС, необхідно зазначити певні технічні вимоги щодо розробки наступної. Насамперед, до таких вимог відносяться:

1) можливість роботи з обладнанням різних виробників;
2) портативність обладнання системи;
3) енергоефективність роботи системи;
4) можливість експорту даних для їх подальшого використання у технічному аналізі режимів роботи станції;
5) доступність використаних матеріалів на ринку України;
6) простота у використанні інтерфейсу програми для можливості роботи з ним персоналу без спеціальної підготовки;
7) відносно невелика вартість компонентів системи.

Для забезпечення належного виконання поставлених вимог, розроблена модель складається з чотирьох функціональних блоків [7], що дозволило спростити як процес розробки системи, так і подальшу експлуатацію окремих її елементів, дає змогу виконувати ремонт обладнання не порушуючи функціювання системи в цілому та легко розширювати існуючий функціонал системи, шляхом інтеграції в неї нових блоків. Отже, розглянемо дану модель більш детально.

Перший блок – блок метеоспостережень, розташований безпосередньо на об’єкті дослідження і призначений для зчитування даних з метеодатчиків та їх передачі по радіозв’язку в блок обробки інформації. Другий блок є блоком зняття електричних даних, здійснює регулярні опитування інвертора для зняття

електричних показників станції й аналогічно першому транслює їх до третього блоку системи. До функціоналу третього блоку, що є блоком обробки інформації, відноситься отримання пакетів даних від фізично віддалених об’єктів дослідження, їх ідентифікації, первинної обробки та подальшої передачі на сервер. Четвертий блок – блок програмного забезпечення, що включає в себе REST-сервер, базу даних та клієнтський додаток.

2.2. Розробка блоку метеоспостережень

В основі роботи блоку метеоспостережень приймаємо мікроконтролер ATMega328, що обумовлено простотою використання та доступністю даного типу контролеру. Характеристики розглянутого мікроконтролера наведені у табл. 2.2.1. Для спрощення експлуатації мікроконтролера ATMega328 в розробленій схемі використовуються готові друковані плати марки Arduino серії Nano (рис. 2.2.1.).

Таблиця 2.2.1 – Характеристики мікроконтролера ATMega328

Параметр	**Значення**	**Параметр**	**Значення**
Тактова частота	0-20 МГц	Flash-пам'ять	32 кб
SRAM-пам’ять	2 кб	EEPROM-пам’ять	1 кб
Напруга живлення	1,8-5,5 В	Струм в режимі роботи	0,2 мА
Загальна кількість портів	23	Струм в режимі сну	0,75 мкА
Кількість ШИМ виходів	6	Кількість каналів АЦП	6
Кількість апаратних USART	1	Кількість апаратних SPI	1

Як вже зазначалося вище, блок метеоспостережень розташований безпосередньо на об’єкті дослідження, тобто фотоелектричного модуля, і призначений для зчитування даних з метеодатчиків та передачі отриманої інформації по радіозв’язку в блок обробки інформації. Він складається з декількох цифрових метеодатчиків, плати Arduino Nano та радіомодуля nRF24L01. Цикл роботи даного блоку умовно можна поділити на декілька етапів:

1) дані, що знімають датчики, оцифровуються та передаються на мікроконтролер з певним інтервалом;
2) мікроконтролер формує пакет з отриманих даних, підписує його та передає на радіомодуль;
3) радіомодуль транслює пакети даних у зарезервований канал.

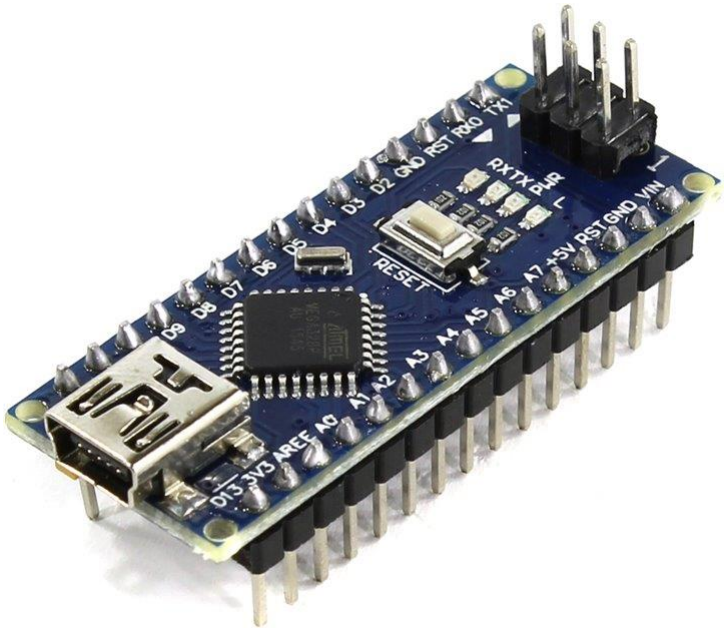
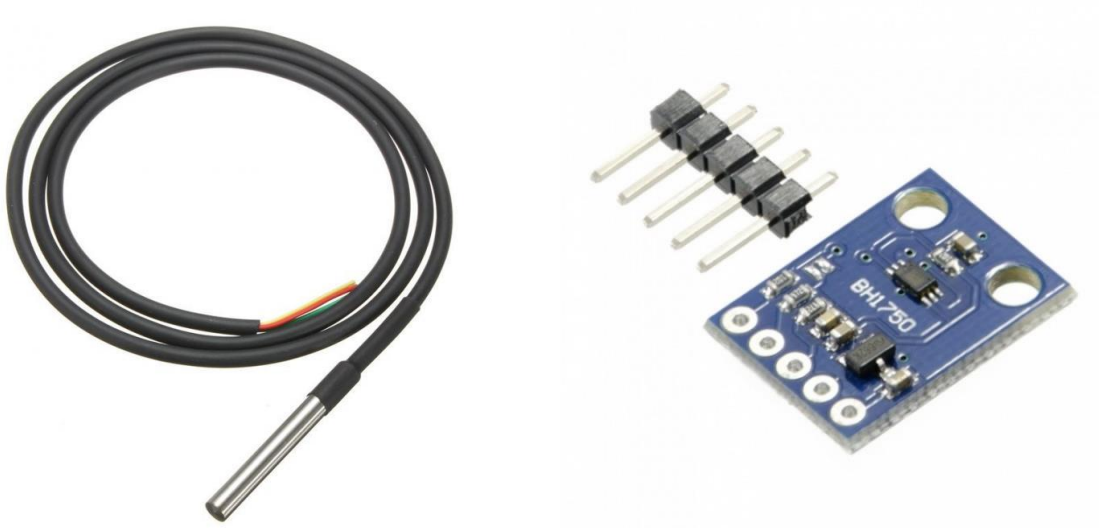


Рисунок 2.2.1 – Друкована плата марки Arduino серії Nano

Отже, у першому етапі циклу роботи блоку метеоспостережень, взаємодія йде між цифровими метеодатчиками та мікроконтролером ATMega328. До метеодатчиків, що наявні у базовій комплектації розробленої системи,

відносяться: цифровий датчик температури DS18B20 (рис. 2.2.2. – а) та цифровий датчик освітленості GY-302 (рис. 2.2.2. – б). Обидва датчики встановлюються безпосередньо на поверхні фотомодуля і взаємодіють з мікроконтролером за визначеним алгоритмом. Передача даних з датчика освітленості виконується за допомогою інтерфейсу I^2C, що є послідовною шиною даних для зв'язку інтегральних схем. В той час, як передача даних з датчика температури виконується через двонапрямлену шину зв’язку – публічний інтерфейс 1-Wire.



а) б)

Рисунок 2.2.2 – Метеодатчики: а) цифровий датчик температури DS18B20; б) цифровий датчик освітленості GY-302

У другому етапі мікроконтролер спочатку виконує первинну обробку інформаційних потоків, тобто формує пакети даних: ідентифікує значення з кожного датчика, підписує їх відміткою часу та здійснює кодування пакету у зручний для передачі формат. Далі за допомогою шини SPI, що по суті є синхронним послідовним повнодуплексним стандартом передачі даних, виконується відправка підготовлених пакетів до радіомодуля nRF24L01, який зображений на рис. 2.2.3.

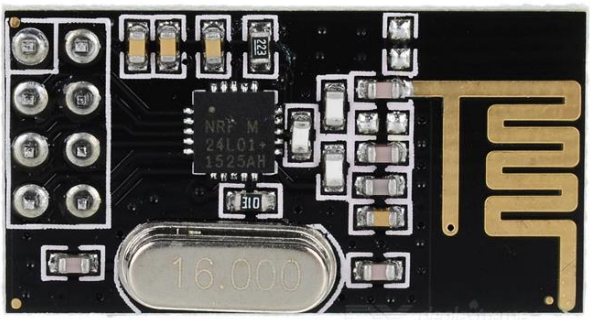


Рисунок 2.2.3 – Радіомодуль nRF24L01

Третій етап є завершальним етапом циклу роботи блоку метеоспостережень і реалізовує зв’язок останнього з блоком обробки інформації. В ході роботи даного етапу радіомодуль nRF24L01 транслює отримані пакети даних у заздалегідь зарезервований канал. Живлення радіомодуля виконується з п’яти вольтового виходу плати Arduino Nano через регулятор напруги 3,3 В AMS1117.

Схематичне зображення описаної вище схеми наведене на рис. 2.2.4. Код прошивки мікроконтролера наведений у додатку 1.

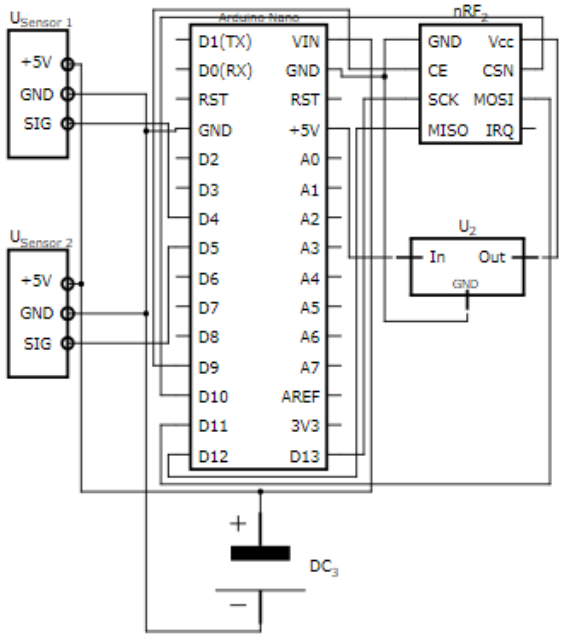


Рисунок 2.2.3 – Принципова схеми блоку метеоспостережень

2.3. Розробка блоку зняття електричних даних

Зібранні моніторинговою системою дані можна умовно поділити на метеорологічні та електричні. До електричних даних відносяться сила струму, напруга, активна та реактивна потужності, частота. Для інтеграції системи збору електричних даних у запропоновану моніторингову систему існує два основні шляхи:

1) використання зовнішньої гальванічної розв'язки з власними вимірювальними пристроями, з подальшою інтеграцією їх вимірювань у розроблену систему;
2) використання вбудованої в інвертор системи моніторингу електричних показників.

Обидва із запропонованих варіантів мають як свої переваги, так і недоліки. Перевагами першого варіанта є не прив'язаність до конкретного бренду/виробника інверторів, можливість зекономити на виборі інвертора, можливість передачі даних за допомогою широкого вибору протоколів зв'язку. До його недоліків можна віднести: складність установки та обслуговування даного обладнання, необхідність побудови власних транзитних ліній передачі даних та електричного живлення обладнання, вимоги до метрологічної точності кожного використаного вимірювального приладу окремо. Перевагами другого варіанта є певна метрологічна точність виміряних даних (інвертор обов'язково підлягає метрологічній оцінці), простота установки та обслуговування. До його недоліків можна віднести: обмеження щодо використання тільки з конкретним брендом/виробником інверторів, дороговизна, обмежена кількість протоколів зв'язку, що підтримуються.

Отже, розглянувши тенденції вибору електричного обладнання ФЕС на сучасному ринку України та враховуючи всі переваги та недоліки вищезгаданих варіантів реалізації системи збору електричних даних щодо роботи ФЕС у ході

експертної оцінки було обрано другий варіант, тобто використання вбудованої в інвертор системи моніторингу електричних показників. Розглянувши ряд представлених інверторів, що поставляються дистриб'юторами фотоелектричного обладнання на ринок України та мають у наявності можливість інтеграції вбудованої системи моніторингу, для подальшого розгляду обираємо інвертор бренду Fronius серії ECO 25.0-3-S (рис. 2.3.1.). Даний вибір обумовлений, насамперед, популярністю виробника, задовільним відношення ціни до якості обраного обладнання, стандартизованого набору функціональності, що представляє інтегрована система моніторингу показників інвертора, що спрощує подальшу інтеграцію систем інших виробників. Параметри інвертора Fronius ECO 25.0-3-S наведені у таблиці 2.3.1 [8].

Рисунок 2.3.1 – Інвертор Fronius ECO 25.0-3-S

Таблиця 2.3.1 – Номінальні параметри інвертора Fronius ECO 25.0-3-S

Параметр	**Значення**
Вага	35,7 кг
Розміри	72.5 x 51 x 22.5 см

Продовження таблиці 2.3.1.

1	2
Номінальна потужність	27 кВт
Кількість фаз	3
Кількість MPPT виходів	1
Інтерфейси підключення	RS-485, WiFI, Ethernet
Ступінь захисту від вологи	IP 66

До даної моделі інвертора обираємо наступну внутрішню моніторингову систему – Fronius Datamaneger 2.0 (рис. 2.3.2.). Fronius Datamanager 2.0 – мережевий реєстратор даних, який поєднує у собі функціональність Fronius Com Card, реєстратора даних Fronius Web, Fronius Power Control Card і Fronius Modbus Card. Даний реєстратор має вигляд друкованої плати та монтується усередині інвертора.

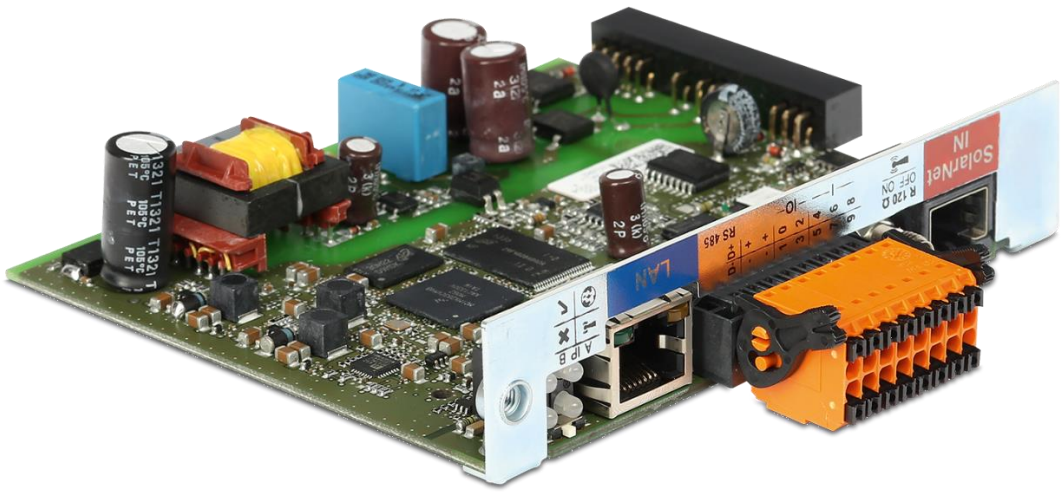


Рисунок 2.3.2 – Мережевий реєстратор даних Fronius Datamanager 2.0

Встановивши даний реєстратор ми маємо можливість використовувати будь-який з трьох інтерфейсів зв’язку:

1) RS-485 (англ. Recommended Standard 485), EIA-485 (англ. Electronic Industries Alliance 485) – стандарт передачі даних двопровідним напівдуплексним багатоточковим послідовним каналом зв'язку. У стандарті для передачі і прийому даних часто використовується одна і та ж вита пара дротів. Передача даних здійснюється за допомогою диференціальних сигналів. Різниця напруги однієї полярності між провідниками означає логічну одиницю, різниця іншої полярності – нуль. Стандарт не нормує формат інформаційних кадрів і протокол обміну. Найчастіше для передачі байтів даних використовуються ті ж фрейми, що і в інтерфейсі RS-232: стартовий біт, біти даних, біт паритету (якщо потрібно), стоповий біт.
2) Ethernet – сімейство протоколів стандарту IEEE 802.3 – це найпопулярніший стандарт серед кабельних комп'ютерних мереж, що працюють на фізичному та канальному рівні мережевої моделі OSI. Ethernet тісно пов'язаний з моделлю TCP/IP, оскільки у переважній більшості випадків служить для передачі IP-пакетів.
3) WiFi – загальновживана назва для стандарту IEEE 802.11 передачі цифрових потоків даних по радіоканалах. Дальність передавання інформації залежить від потужності передавача (яка в окремих моделях обладнання регулюються програмно), наявності та характеристики перешкод, типу антени.

Мережевий реєстратор Fronius Datamanager 2.0 має можливість передавати дані по кожному з приведених вище інтерфейсів використовуючи наступні технології (рис. 2.3.3.): Modbus (TCP, RTU), HTTP API, push service (HTTP, FTP).

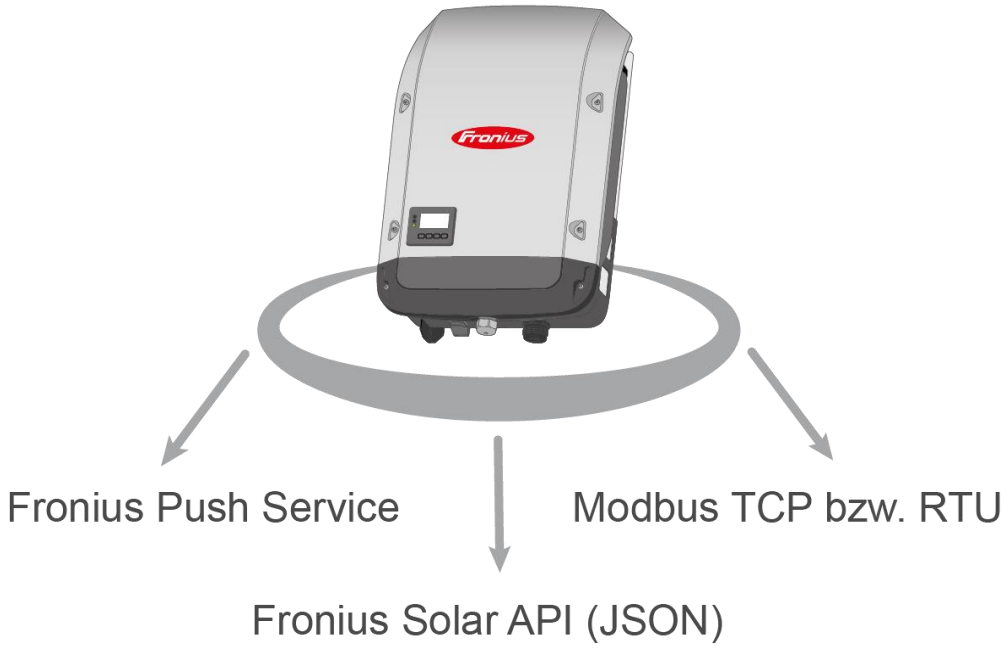


Рисунок 2.3.3 – Протоколи передачі даних мережевим реєстратором Fronius Datamanager 2.0

Для зручності інтеграції даного мережевого реєстратора у розроблену моніторингову систему скористаємося публічним інтерфейсом Ethernet для отримання даних за допомогою HTTP API, що реалізоване по типу REST-сервісу. REST (англ. Representational State Transfer) – підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів, в основі якого закладені умови кешування даних, не залежність від мережевого прошарку та відсутність зберігання інформації про стан між парами «запит-відповідь». В даний час єдиним способом взаємодії з цим API є створення HTTP-запиту для певного CGI. Реалізоване API у Fronius Datamaneger 2.0 практично розділяє запити на два види: запити даних знятих в реальному часі та запити архівних даних. Запити даних в реальному часі отримуються безпосередньо з внутрішніх пристроїв інвертора, тому їх можна використовувати тільки тоді, коли інвертор підключений та не знаходиться у стані очікування. Зняті дані передаються у форматі JSON, що по суті є текстовим форматом обміну даними.

Так як друковані плати марки Arduino, що використовуються у запропонованій моніторинговій системі, не мають вбудованого інтерфейсу Ethernet, який необхідний для обраного способу підключення мережевого реєстратора Fronius Datamanager 2.0, скористаємося розширювальною платою W5100 Ethernet Shield (рис. 2.3.4.). Спілкування між Arduino і розширювальною платою побудовано за допомогою шини SPI, послідовному периферійному інтерфейсі, що є синхронним послідовним повнодуплексним стандартом передачі даних.

Отже, до технічної бази виконання блоку зняття електричних даних ФЕС входять: друкована плата марки Arduino серії UNO (спільна для блоку зняття електричних даних і блоку обробки інформації), розширювальна плата W5100 Ethernet Shield та мережевий реєстратор Fronius Datamanager 2.0, інтегрований у інвертор Fronius ECO 25.0-3-S. Розглянемо алгоритм взаємодії між ними.

З певним інтервалом часу Arduino формує HTTP-запит до Fronius Datamanager 2.0 за даними занятими з інвертора у реальному часі. Запити до HTTP API потребують наявності веб-клієнта, що у реалізованій схемі будується на базі Arduino, з використанням стандартної бібліотеки «Ethernet.h» та плати W5100 Ethernet Shield, що по суті відіграє роль модулятора у даній схемі. Так, як HTTP API запропоноване в обраному мережевому реєстраторі реалізоване по типу локального REST-сервера, для комунікації з ним використовуються звичайні GET HTTP/1.1 запити. Усі публічні методи, що підтримує даний мережевий реєстратор наведені у документації до нього – «Fronius Solar API V1». Наведемо приклад запиту, що має повернути дані зняті з внутрішніх вимірювальних приладів інвертора у реальному часі [9]:

```
GET /solar_api/v1/GetMeterRealtimeData.cgi HTTP/1.1
Host: 169.254.0.180
```

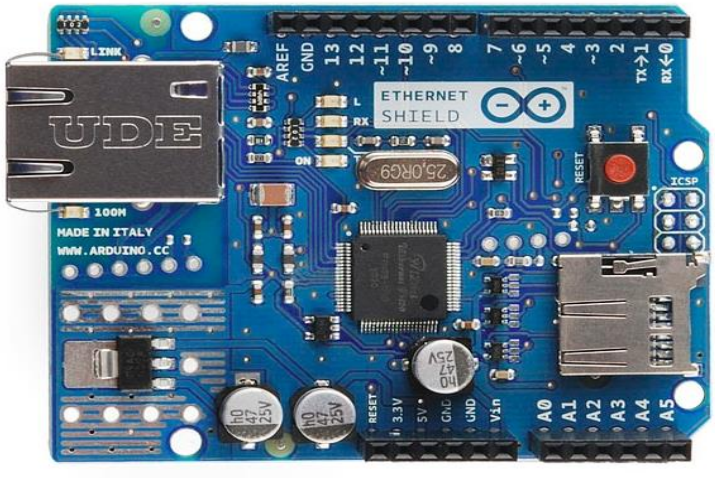


Рисунок 2.3.4 – Розширювальна плата W5100 Ethernet Shield

У відповідь на даний запит мережевий реєстратор формує і надсилає JSON файл, у якому наведені наступні параметри:

1) відмітка часу коли був отриманий запит;
2) дійсне значення величини активної потужності кожної фази;
3) дійсне значення сумарної активної потужності;
4) дійсне значення величини реактивної потужності кожної фази;
5) дійсне значення сумарної реактивної потужності;
6) дійсне значення величини змінного струму кожної фази;
7) дійсне значення величини змінної напруги кожної фази;
8) дійсне значення величини змінної напруги між фазами (AB, BC, CA);
9) дійсне значення коефіцієнта потужності кожної фази.

Після декодування та ідентифікації отриманих даних, мікроконтролер передає їх у блок обробки інформації.

Схематичне зображення описаної вище схеми наведене на рисунку 2.3.4. Код прошивки мікроконтролера наведений у додатку 2.

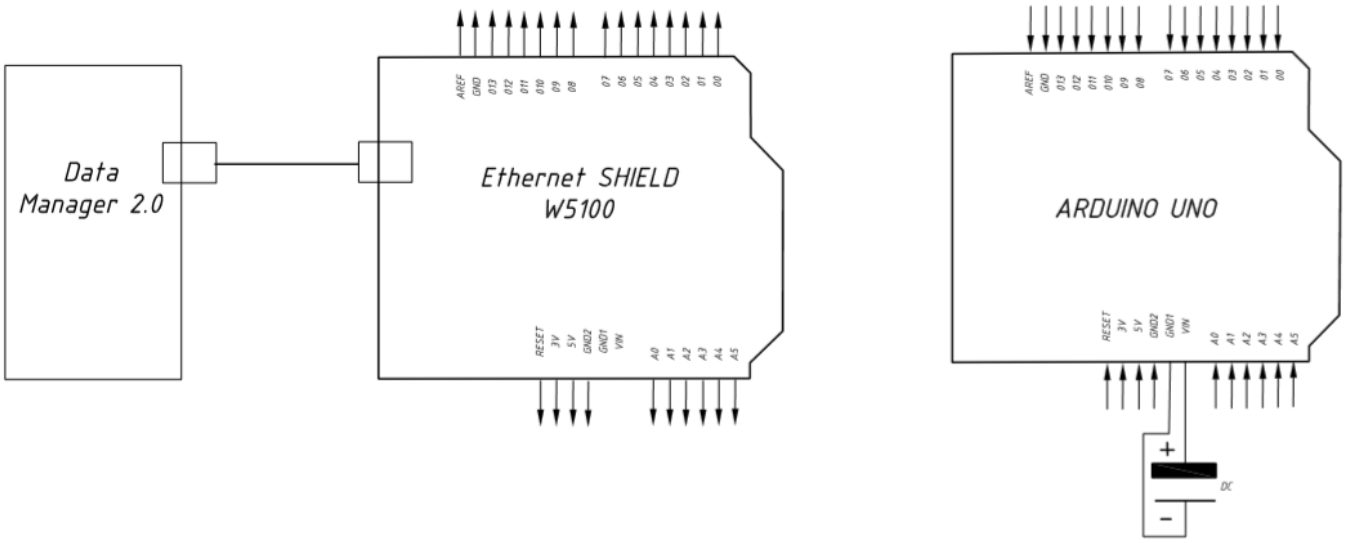


Рисунок 2.3.4 – Принципова схема блоку збору електричних даних

2.4. Розробка блоку обробки інформації

Блок обробки інформації є одним з основних блоків розробленої системи, забезпечує можливість масштабування, надаючи змогу легко інтегрувати в систему як нові датчики, так і нові об’єкти дослідження (наприклад, акумуляторні батареї). До основної функціональності даного блоку належать:

1) збір даних – отримання пакетів даних від фізично віддалених об’єктів дослідження;
2) ідентифікація пакетів даних – визначення до якого параметру системи належать ті чи інші отримані показники;
3) первина обробка даних – приведення отриманих з різних віддалених датчиків показників до одних одиниць міжнародної системи СІ;
4) перекодування інформації – переведення отриманих пакетів даних у зручний для подальшої передачі формат;
5) передача інформації – подальша передача даних на логуючий сервер.

Аналогічно блоку метеоспостережень в основі роботи блоку обробки інформації приймаємо мікроконтролер ATMega328 з єдиною відмінністю у тому, що замість друкованої плати Arduino Nano використаємо друковану плату Arduino Uno (рис. 2.4.1.). Дане рішення обумовлено більшою кількістю цифрових та аналогових входів та виходів, більшою оперативною пам’яттю мікроконтролера, що використаний у даній серії плат.

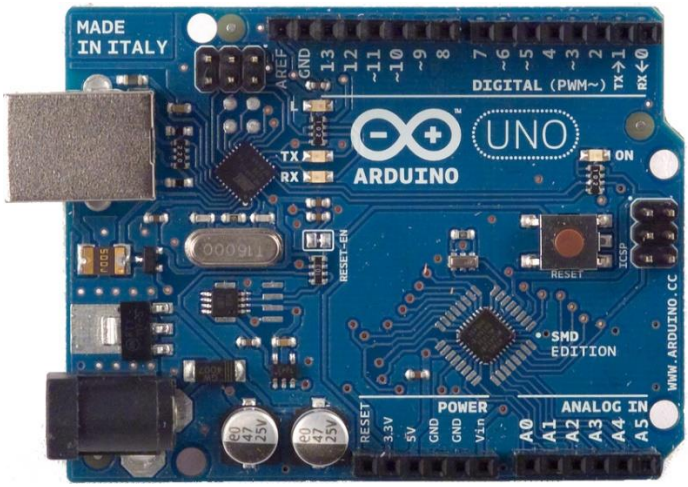


Рисунок 2.4.1 – Друкована плата марки Arduino серії Uno

Для забезпечення зв’язку з фізично віддаленими об’єктами дослідження, як вже вказувалося раніше, використовується радіомодуль nRF24L01, в той час як забезпечення зв’язку із сервером виконується за допомогою GSM-модуля на базі чіпа фірми SIMCom Wireless Solutions серії SIM800c (рис. 2.4.2.). Варто зауважити, що так як блок збору електричних даних і блок обробки інформації використовують спільний мікроконтролер, немає необхідності будувати канал зв’язку для отримання знятих даних щодо електричних параметрів роботи системи. Розглянутий блок використовує два публічні інтерфейси зв’язку:

1) SPI – використовується для зв’язку з радіомодулем nRF24L01;
2) UART – для використовується для зв’язку з GSM-модулем SIM800c.

Рисунок 2.4.1 – GSM-модуль SIM800c

Так як використана у блоці зняття електричних параметрів розширювальна плата W5100 Ethernet Shield для зв’язку з мікроконтролером також використовує послідовний інтерфейс SPI, для спільного використання даного інтерфейсу з радіомодулем nRF24L01 існує два варіанти: використання можливості апаратного переривання, чи використання програмного аналогу SPI шині, тобто побудови даного інтерфейсу на будь-яких інших виходах Arduino окрім тих, що приймають участь у апаратному SPI. З точки зору ефективності використання мікроконтролера перший варіант є більш оптимальним, але в той же час з точки зору швидкості розробки даний варіант дуже сильно програє другому. Отже, так як для підтримки обраних публічних інтерфейсів зв’язку кількості виходів Arduino Uno предостатньо, зупинимось на останньому варіанті. Для його реалізації скористаємося стандартною бібліотекою «SoftSPI.h», що входить у пакет Arduino IDE. Також, так як код прошивки мікроконтролера для комунікації з радіомодулем використовує бібліотеку «RF24.h», необхідно внести до неї певні зміни, адже дана бібліотека зав’язана на використання апаратного SPI. Усі зміни вихідного коду бібліотеки «RF24.h» наведені у додатку 5.

Для зв’язку блоку обробки інформації з сервером, як вже вказувалося раніше, використовується GSM-модуль SIM800c. Даний модуль має деякі

особливості стосовно його експлуатації. Так як робоча напруга даного приладу становить 4.2-4.5 В, а споживаний струм у пікові моменти роботи може досягати значення у 2 А – це робить неможливим його живлення від Arduino, яка у свою чергу живиться напряму від логічного виходу плати Fronius Datamanager 2.0 вбудованої в інвертор. Тому для забезпечення належного живлення даного приладу, використаємо наступну схему підключення модуля:

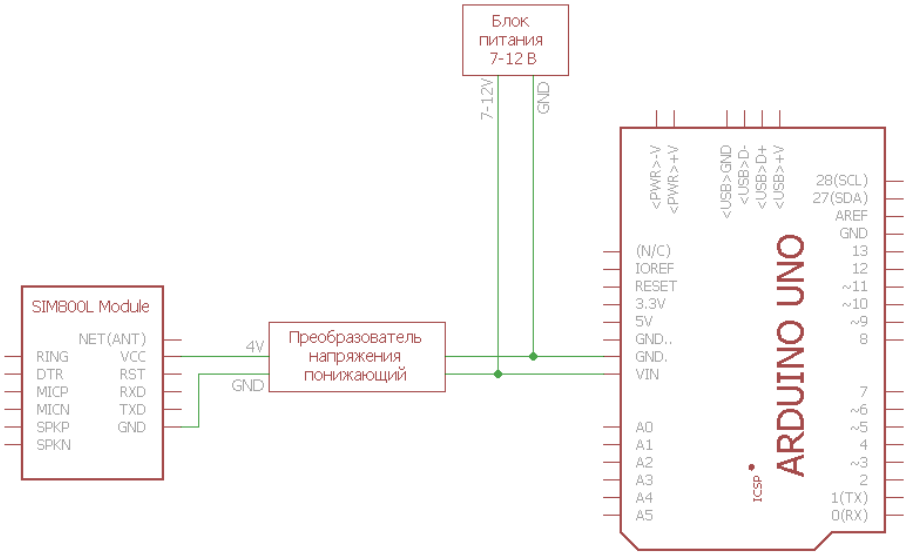


Рисунок 2.4.2 – Схема живлення GSM-модуля SIM800c

У якості перетворювача використаємо понижуючий конвертор напруги MP1584 (рис. 2.4.3.). Характеристики даного приладу наведені в таблиці 2.4.1.

Таблиця 2.4.1 – Характеристики понижуючого конвертора напруги MP1584

Параметр	**Значення**
Вхідний діапазон напруг	4.5 - 28 В
Вихідний діапазон напруг	0.8 - 20 В
Максимальний вихідний струм	3 А
Коефіцієнт корисної дії	до 96%

Продовження таблиці 2.4.1.

1	2
Пульсації на виході	до 30 мВ
Частота переключення	1 МГц
Робочий діапазон температур	-45-85 °C
Розміри	22 х 17 х 4 мм

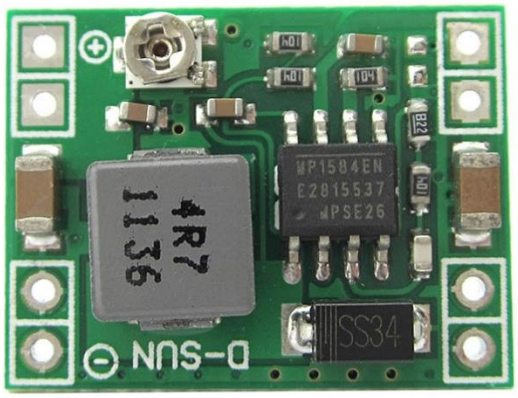


Рисунок 2.4.3 – Понижуючий конвертор напруги MP1584

Отже, щойно була описана апаратна структура блоку обробки інформації, тепер розглянемо алгоритм роботи системи. Першим етапом циклу роботи блоку обробки інформації є отримання пакетів даних з фізично віддалених об’єктів дослідження – передача даних з блоку метеоспостережень виконується по радіозв’язку, передача даних з блоку зняття електричних параметрів виконується через SPI-шину. Далі система перевіряє ідентифікатори пакетів даних, що повинні відповідати ідентифікаторам датчиків у системі. Наступним етапом циклу роботи системи є підготовка інформації до передачі на сервер. Вході підготовки виконуються наступні дії: приведення ідентифікованих показників до відповідних одиниць міжнародної системи СІ, перекодування пакетів даних у формат JSON – текстовий формат передачі даних. Завершальним етапом роботи

циклу блока обробки інформації є передача підготовленого JSON-рядка на сервер за допомогою POST HTTP/1.1 запиту.

Схематичне зображення описаної вище схеми наведене на рис. 2.4.4. Код прошивки мікроконтролера наведений у додатку З.

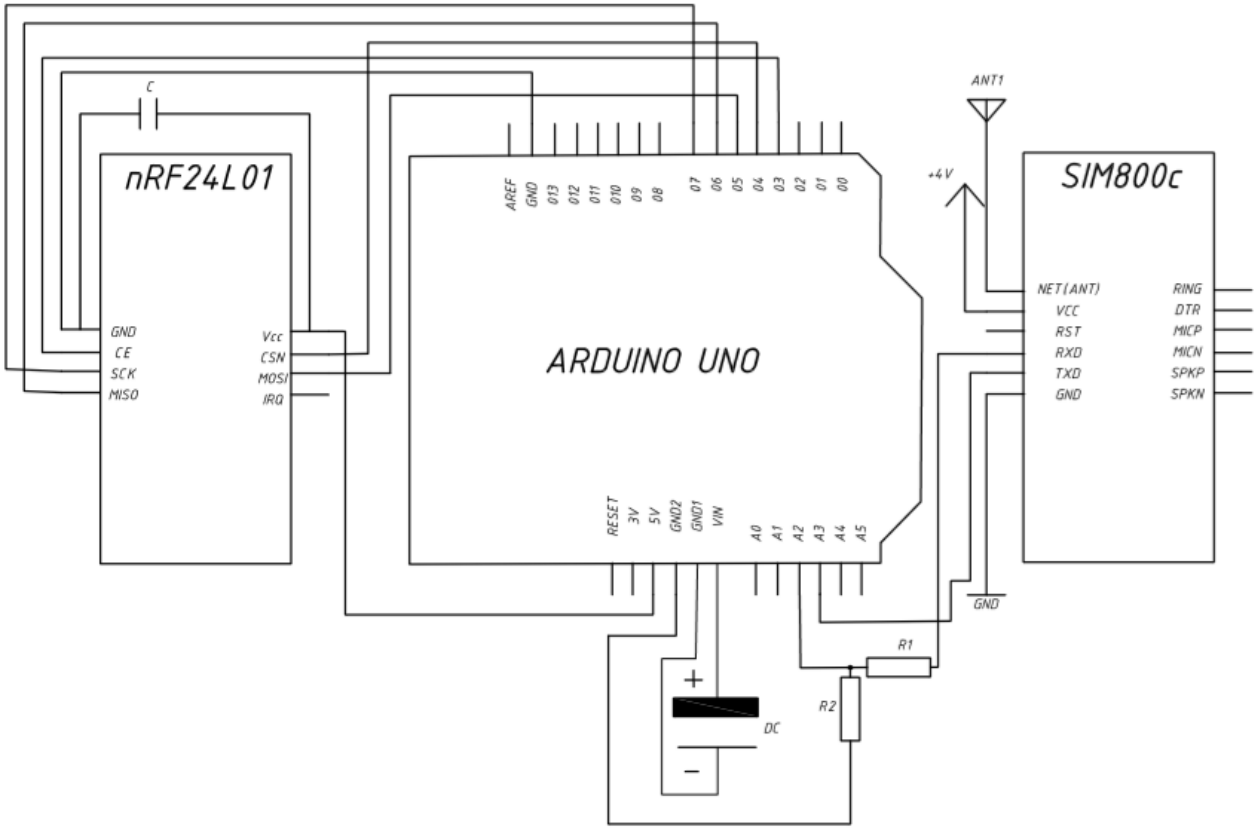


Рисунок 2.4.4 – Принципова схема блоку обробки інформації

2.5. Розробка блоку програмного забезпечення

Виходячи із умов надійності та масштабованості системи, можливого розповсюдженого використання розробленого програмного забезпечення та можливої інтеграції нових блоків у розглянуту модель для розробки останнього блоку системи обираємо архітектурне рішення – клієнт-серверного додатку. Архітектура клієнт-сервер (рис. 2.5.1.) є одним із структурних шаблонів проектування програмного забезпечення та є домінуючою концепцією у

створенні розподілених мережевих застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти [10]:

1) набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
2) набір клієнтів, які використовують сервіси, що надаються серверами;
3) набір баз даних, що впроваджують зберігання, резервне копіювання та надання по запиту інформації.

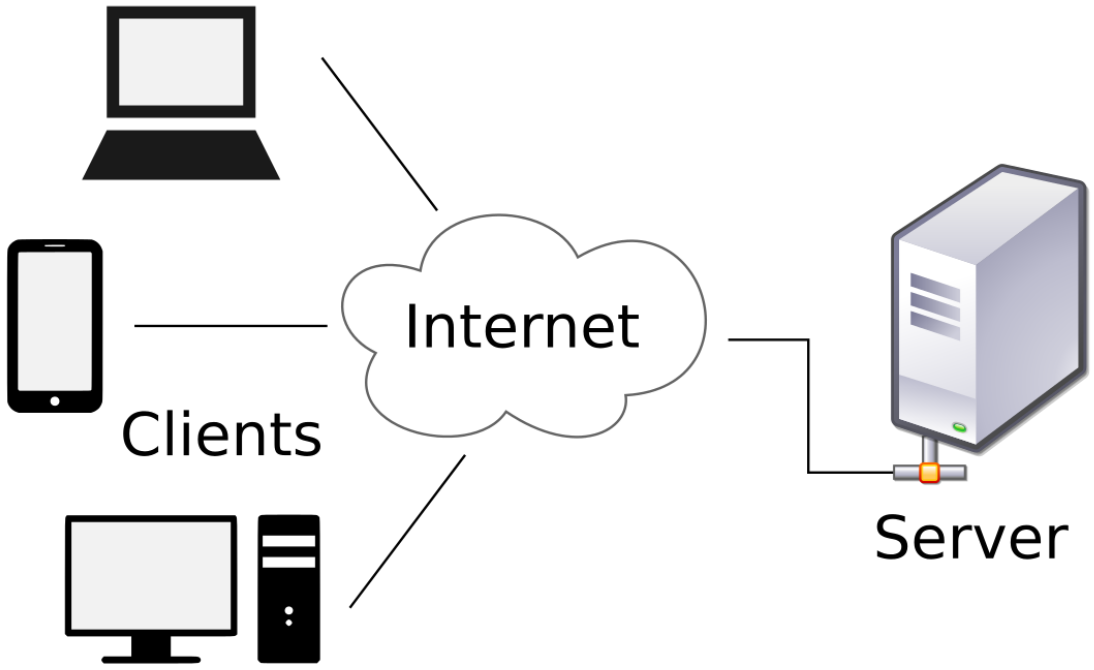


Рисунок 2.5.1 – Схематичне зображення клієнт-серверної архітектури

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Для розробки блоку програмного забезпечення зупинимось на наступному варіанті клієнт-серверного додатку – кожен клієнт знає лише про один доступний балансуючий сервер, надсилаючи на нього запити, сервер активує балансуючий пристрій, що переадресовує запит на відповідний серверний додаток. В подальшому розвитку системи, можливе використання кластеризації балансуючих серверів, але ні це, ні балансуючий пристрій у даній дипломній роботі не описуються, так як це є частиною архітектури мережі і не є темою даного дослідження.

Отже, розглянемо основні частини і алгоритм роботи розробленої системи.

Серверний додаток виконується за принципом REST-сервера. Короткий опис даної методології вже приводився раніше, тому не будемо зупинятись на цьому ще раз. Основним стеком технологій, що використовуються у розробленому сервісі, є:

1) Платформа Node.js – платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript.
2) Система керування базами даних PostgreSQL – об'єктно-реляційна система керування базами даних (надалі – СКБД).

Так як мережевий застосунок написаний з урахуванням принципів методології REST-сервісу, сервер не зберігає стан між парами запит-відповідь. Тому алгоритм роботи сервера можна представити як набір обробників запитів (рис. 2.5.2.). Тобто, кожен HTTP/1.1 запит до розробленого сервісу проходить через низку обробників, так званих «middleware», кожен з яких може як декорувати отриманий на вході запит – проміжний обробник, так і перервати ланцюг відповівши на запит – кінцевий обробник.

Рисунок 2.5.2 – Блок схема алгоритму роботи серверного додатку

До проміжних обробників запитів, що наявні у розробленій системі, можна віднести наступні ланки:

1) ланка декодування HTTP-cookie параметрів – обумовлено потребою збору інформації щодо статистики використання розробленого програмного забезпечення;
2) ланка декодування HTTP-body параметрів – обумовлено потребою обробки корисного навантаження запиту, так званий «payload».

До кінцевих обробників запитів відносяться наступні ланки:

1) ланка обробки запиту на завантаження нових значень отриманих з блоку обробки інформації;
2) ланка обробки запиту статистичних даних спостережень.

Розберемо алгоритм роботи кінцевих обробників запитів більш детально.

Отже, коли серверний додаток отримує HTTP-запит на завантаження даних отриманих з блоку обробки інформації, спочатку відбувається первинна підготовка запиту проміжними ланками «middleware», після чого робота переходить до кінцевого обробника запиту, алгоритм роботи якого зображено на рис. 2.5.3.

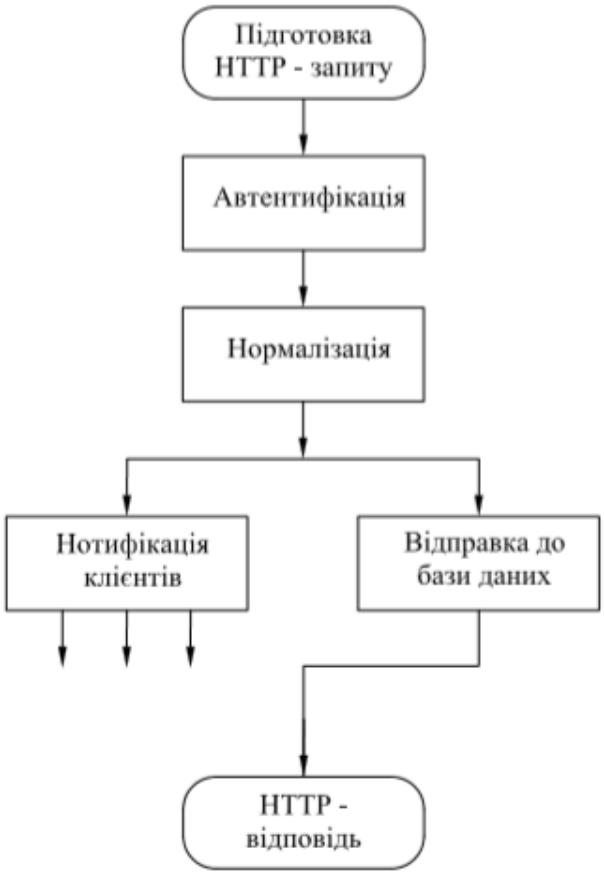


Рисунок 2.5.3 – Блок схема алгоритму роботи ланки обробки запиту на завантаження нових значень

Ключовими етапами циклу роботи вищенаведеного обробника запитів є автентифікація отриманих пакетів даних, тобто визначення надійності джерела інформації; нормалізація отриманої інформації, тобто приведення пакетів даних до формату, що відповідає структурі таблиці бази даних; паралельна відправка нормалізованих показників до бази даних та нотифікація усіх активних клієнтських додатків про наявність нових актуальних значень знятих з фізично

віддалених датчиків у реальному часі; формування HTTP-відповіді на запит. Нотифікація активних клієнтських додатків реалізується за допомогою протоколу WebSocket – протокол, призначений для обміну інформацією між клієнтським додатком та веб-сервером в режимі реального часу, забезпечуючи двонаправлений повнодуплексний канал зв'язку через один TCP-сокет.

Аналогічно вище описаному варіанту обробки запиту, коли серверний додаток отримує HTTP-запит за статистичними даними спостережень, після первинної підготовки робота переходить до кінцевого обробника, алгоритм роботи якого зображено на рис. 2.5.4.

Рисунок 2.5.4 – Блок схема алгоритму роботи ланки обробки запиту статистичних даних спостережень

Ключовими етапами циклу роботи приведеного кінцевого обробника запитів є серіалізація query-параметрів для безпечної передачі їх до SQL-скрипту; формування відповідного до query-параметрів запиту до бази даних; формування відповідної HTTP-відповіді на запит.

Для побудови повноцінної екосистеми і впровадження механізму донесення інформації до кінцевого споживача, тобто користувача даного

програмного забезпечення, а також для покращення сприйняття статистичних даних оператором, було прийняте рішення розробити клієнтський додаток у вигляді комп'ютерної програми. Для забезпечення максимальної гнучкості до параметрів персонального комп'ютера (надалі – ПК) і збільшення потенціальної кількості користувачів ПЗ було прийнято рішення розробити кросплатформенний додаток з використанням останніх технологій. Вибір операційних систем (надалі – ОС), що підтримуватимуться розробленим рішенням, робимо основуючись на їх розповсюдженості на сучасному світовому ринку. Отже, додаток повинен підтримувати наступні ОС: OS X, Linux, Windows. Для забезпечення поставлених вимог до розробленого ПЗ обираємо наступний стек технологій:

1) Electron – кросплатформений фреймфорк для створення нативних графічних додатків для настільних ОС за допомогою веб-технологій. Фреймворк включає в себе Node.js, для роботи з backend-частиною, і бібліотеку рендеринга Chromium для роботи с frontend-частиною.
2) React – JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми будування інтерактивних інтерфейсів та динамічного оновлення частин сторінки.

Розроблена програма – це потужний інструмент для дистанційного спостереження та обслуговування станції, що забезпечує швидкий доступ до всіх оперативних даних починаючи від показників виробленої електроенергії до стану обладнання електростанції. Дане рішення об'єднує у собі можливості доступу до значень показників спостереження як у реальному часі, так і статистичних даних, що збирались протягом певного часу, впроваджує функціонал нотифікації обслуговуючого персоналу про можливі неполадки і можливість моніторингу одразу декількох ФЕС.

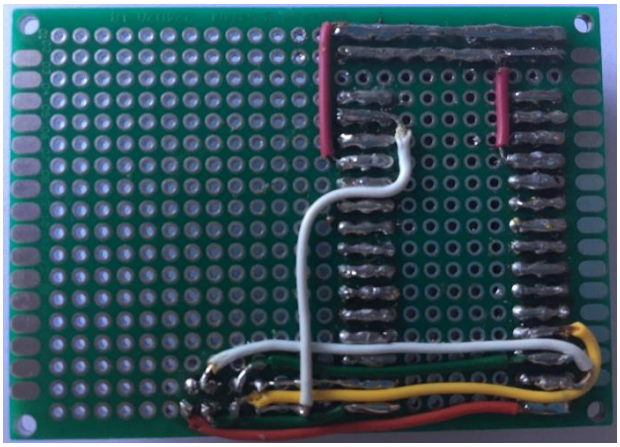
Код розробленого клієнт-серверного застосунку наведений у додатку 4.

2.6. Результати розробки

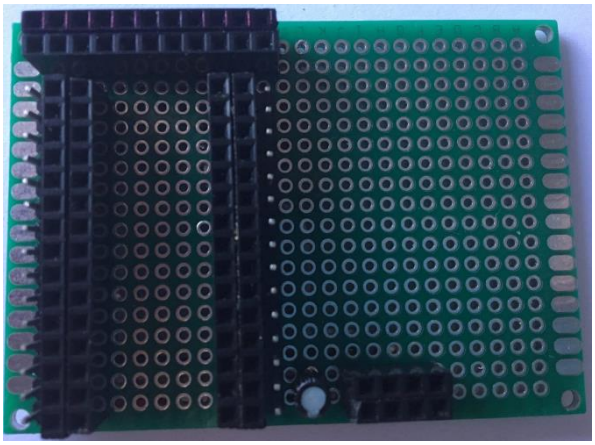
Результатом роботи дипломного проекту є розроблена система моніторингу ФЕС. Дана система виконана з використанням доступних матеріалів, сучасного обладнання та прогресивних технологій. Розробка являє собою цілу екосистему, адже покриває весь цикл роботи, починаючи від збору даних з фізично віддалених датчиків та приладів до програмної обробки інформації з подальшою можливістю додавання технічного аналізу та прогнозування. Розроблену систему можна розглядати з таких двох сторін – апаратна та програмна частини.

Як вже зазначалося раніше, апаратна частина системи складається з метеодатчиків (на прикладі датчика освітленості GY-302 та датчика температури DS18B20), мікроконтролерів ATMega328 в складі друкованих плат Arduino, мережевого реєстратора даних Fronius Datamanager 2.0, інвертора Fronius серії ECO, радіомодулей nRF24L01, розширювальної плати W5100 Ethernet Shield, понижуючого конвертора напруги MP1584 та GSM-модулю SIM800c. Кожний з вищенаведених модулів взаємодіють один з одним, відіграючи свою роль у системі. В результаті виконаних робіт було змонтовано та розпаяно на мекетних платах два основні пристрої: прилад збору показників з метеодатчиків та передачі отриманої інформації по радіозв’язку (блок метеоспостережень) та прилад для опитування інвертора, агрегування інформації та її подальшої передачі на сервер через HTTP/1.1 запити (блоки зняття електричних даних та обробки інформації).

Розглянемо схему підключення компонентів приладу першого. Для підключення радіомодуля до мікроконтролера виконані наступні з’єднання, що зображені на рис. 2.6.1.



а)



б)

Рисунок 2.6.1 – Схема підключення радіомодуля nRF24L01 до друкованої плати Arduino Nano: а) вигляд макетної плати знизу; б) вигляд макетної плати зверху

Як видно на вищенаведених рисунках для узгодження живлення радіомодуля від мікроконтролера використовується конденсатор ємністю 4,7мкФ для згладжування піків споживання напруги. Повна схема підключення метеодатчиків до даного приладу наведена на рис. 2.6.2.

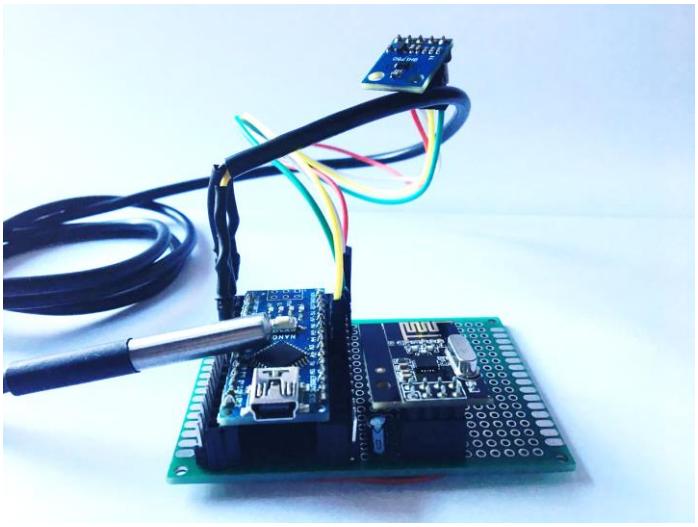
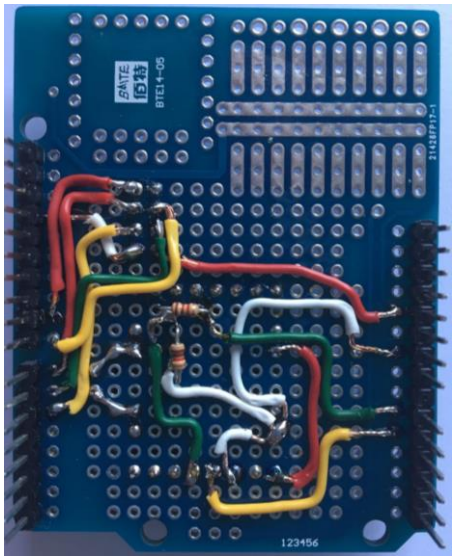
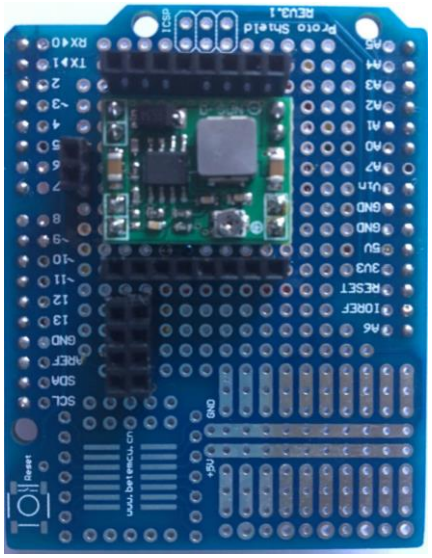


Рисунок 2.6.2 – Схема підключення метеодатчиків до розробленого приладу

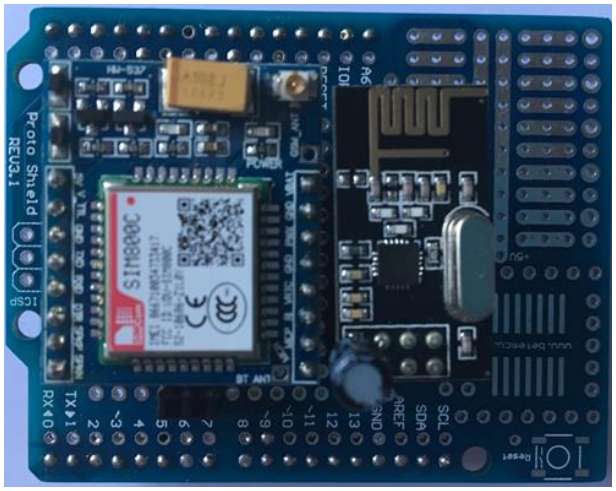
Для підключення комунікуючих компонентів другого приладу було виконано наступну схему, зображену на рисунку 2.6.3. Як було описано в розділі 2.4. для узгодження рівнів робочої напруги мікроконтролера та GSM-модулю в схемі використовується понижуючий конвертор напруги. Аналогічно попередньому приладу паралельно до живлення радіомодуля вмикається конденсатор на 4,7 мкФ.



а)



б)



в)

Рисунок 2.6.3 – Схема підключення радіомодуля nRF24L01 та GSM-модуля SIM800c до Arduino Uno: а) вигляд макетної плати знизу; б) вигляд

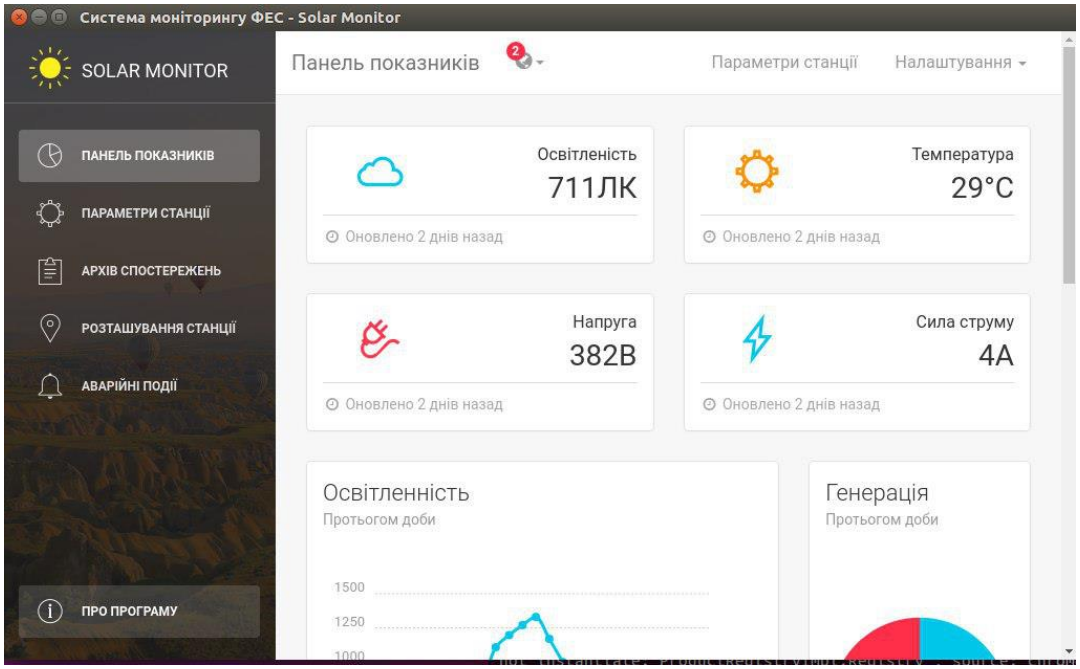
макетної плати зверху; в) вигляд макетної плати зверху з підключеними приладами комунікації

Повна схема підключення другого розробленого приладу з урахуванням підключення до мережевого реєстратора даних Fronius Datamanager 2.0, завчасно підключеного до інвертора Fronius ECO, зображена на малюнку 2.6.4.

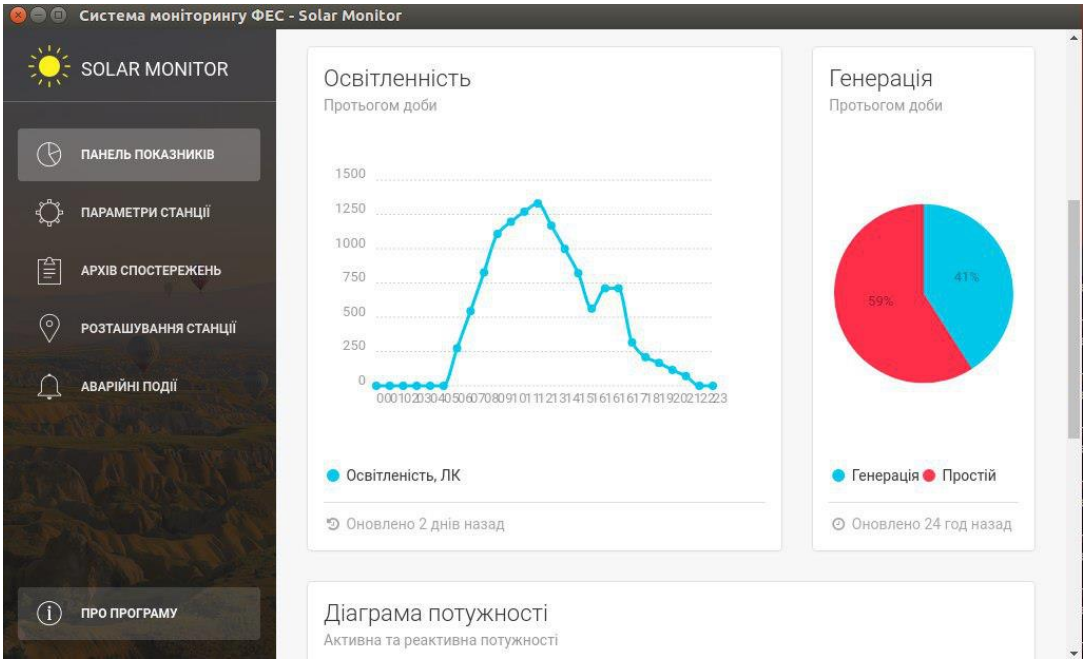
Рисунок 2.6.4 – Підключення спільного приладу блоків зняття електричних даних та обробки інформації до мережевого реєстратора даних Fronius Datamanager 2.0 вбудованого в інвертор Fronius серії ECO

Програмна частина розробленої системи складається з двох основних складових: коду прошивки мікроконтролерів та клієнт-серверного застосунку. Як вже зазначалося раніше програмний код вищезгаданих елементів наведений у відповідних додатках до дипломної роботи. Інтерфейс розробленої програми складається з п’яти екранів. Розберемо кожен із них окремо.

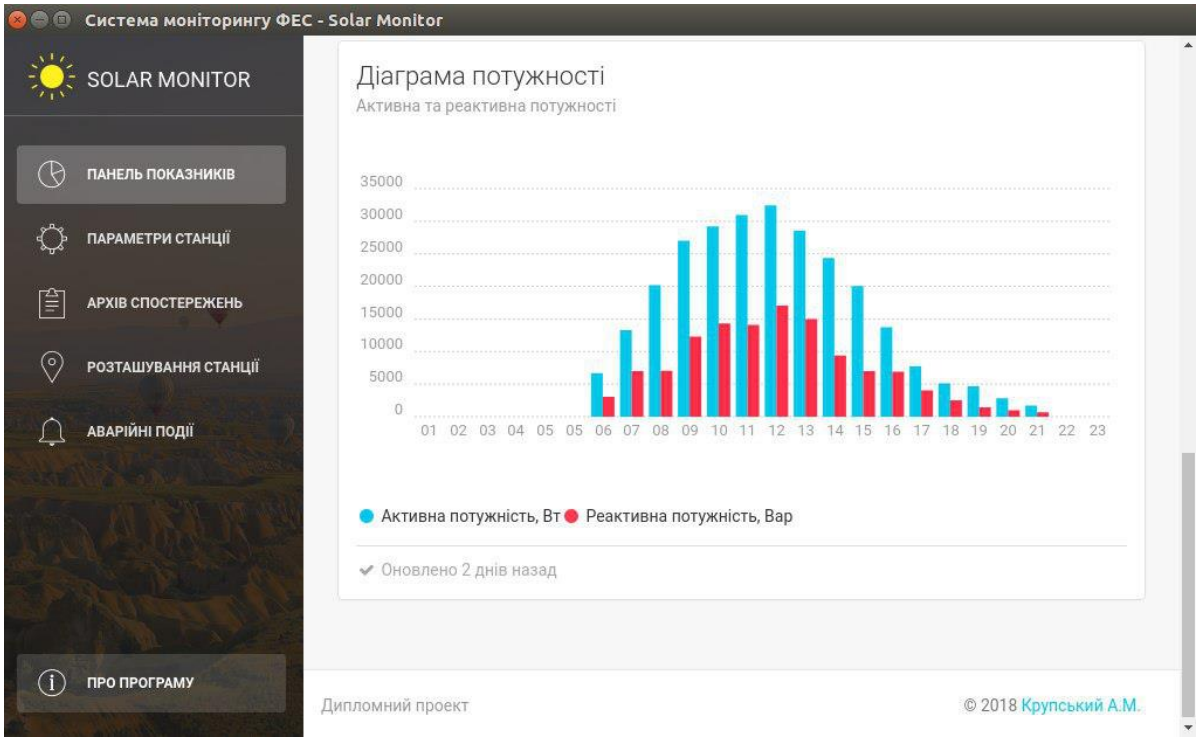
На рис. 2.6.5 зображений стартовий екран розробленої програми. Розглянувши зміст наведеної сторінки можна виділити декілька основних частин: блок навігації по програмі, панель показників реального часу та блок статистичних даних.



а)



б)



в)

Рисунок 2.6.5 – Стартовий екран розробленого застосунку: панель показників реального часу; б) статистичні дані освітленості та генерації; в) діаграма потужності

На другому екрані розробленої програми зображені параметри досліджуваної ФЕС враховуючи встановлене обладнання на станції, їх основні параметри та характеристики. Даний розділ може бути зручним для автоматизації інвентаризації встановленого обладнання та представляє зручний спосіб представлення технічних параметрів ФЕС. Зображення розглянутого екрану наведене на рисунку 2.6.6.

Третій екран несе у собі інформативну роль, так як є кладовищем усіх статистичних даних протягом усього періоду спостережень. На даний момент часу за замовчування програма виводить 100 останніх знятих показників, але у разі необхідності дане число дуже легко змінити – як в меншу, так і в більшу сторону. Вищеописаний інтерфейс зображено на рис. 2.6.7.

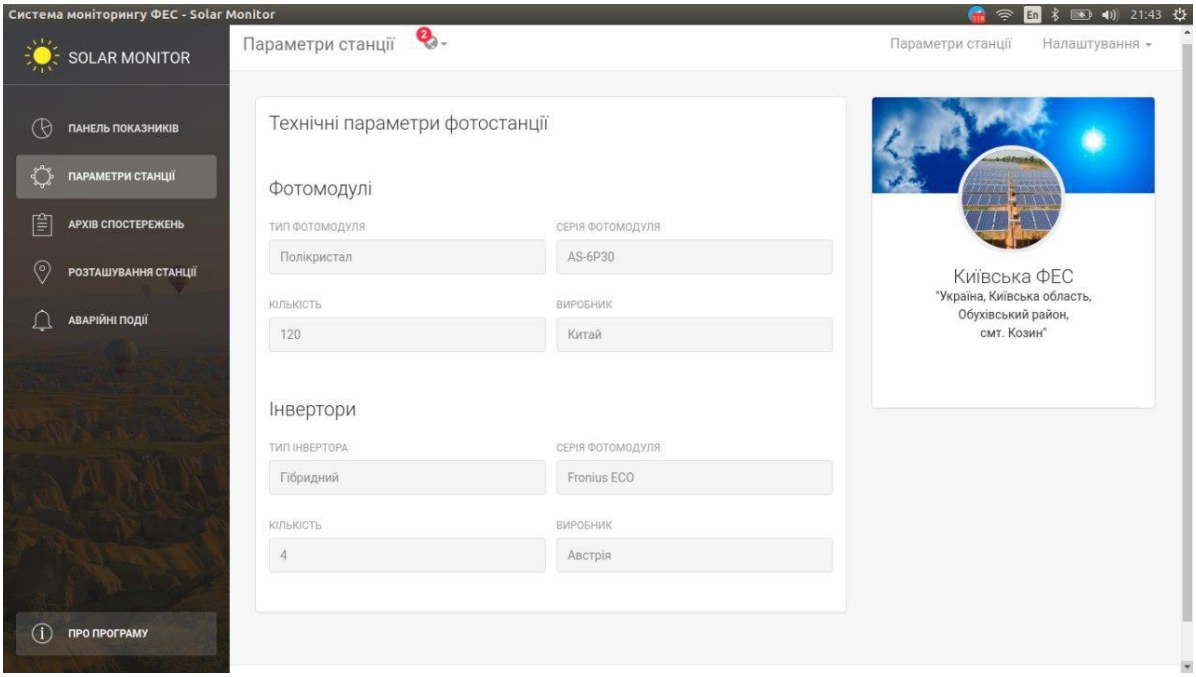


Рисунок 2.6.6 – Другий екран розробленої програми – екран параметрів досліджуваної ФЕС.

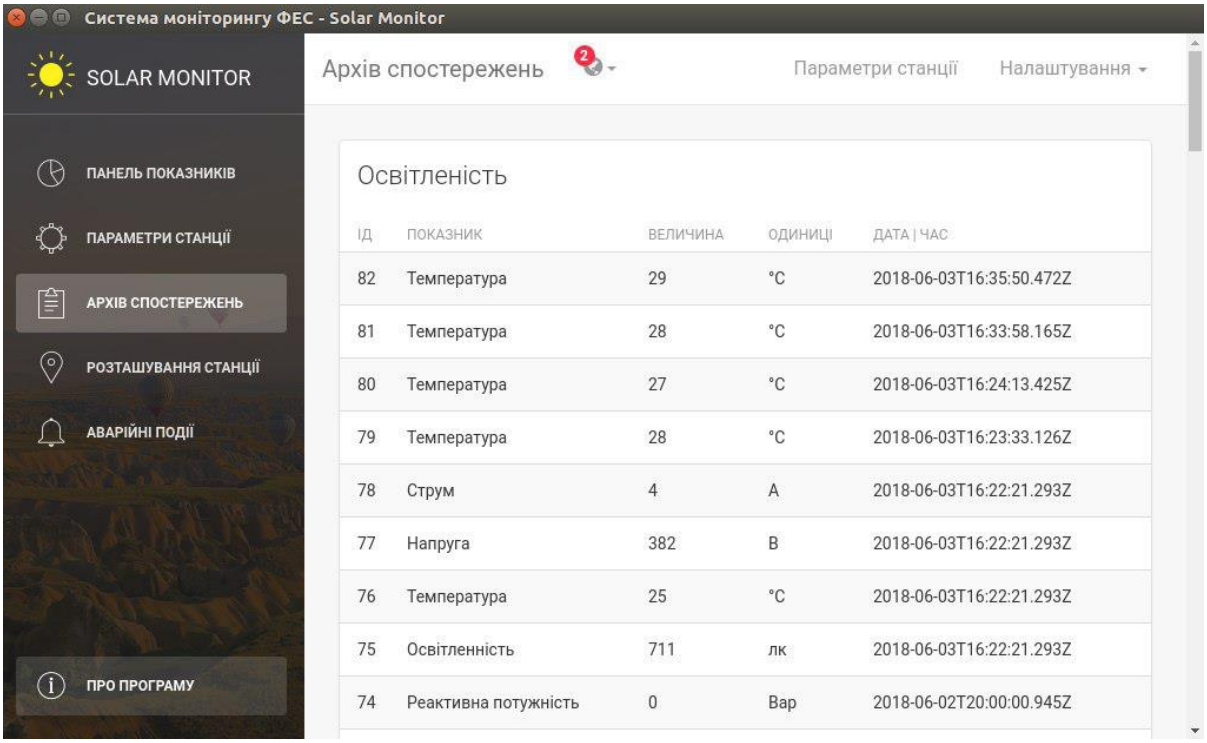


Рисунок 2.6.7 – Третій екран розробленої програми – екран статистичних даних.

У випадку встановлення двох та більше електростанцій пропунується зручний інтерфейс наочного зображення розташування ФЕС, в центрі якого знаходиться інтерактивна мапа, що для відображення використовує публічний інтерфейс Google API. Даний інтерфейс зображений на рис. 2.6.8.

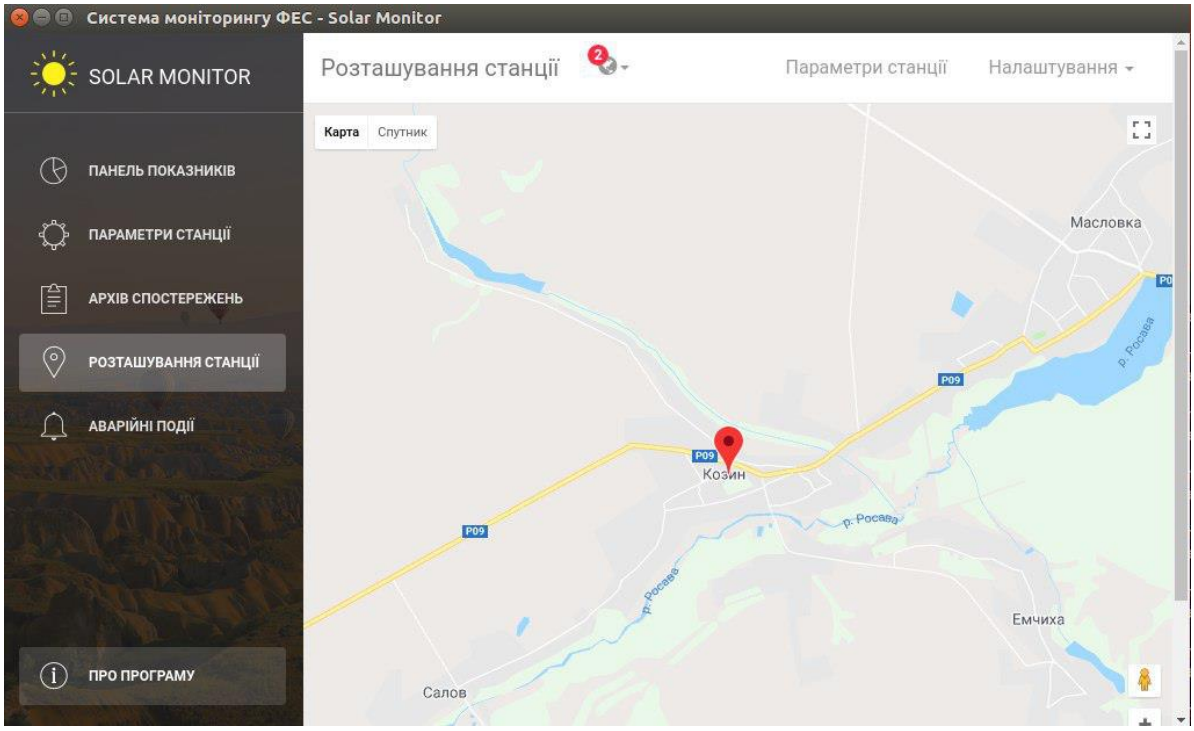


Рисунок 2.6.7 – Четвертий екран розробленої програми – екран розташування електростанцій

Останній за номером, але не за важливістю є п’ятий екран, що призначений для зберігання нотифікацій користувача щодо подій які відбуваються при роботі ФЕС та моніторингової системи включно. Дані повідомлення поділяються на два типи: апаратні та програмні. Апаранті нотифікації призначені для виводу повідомлень про можливі пошкодження та аварійні режими, які можуть виникати під час роботи ФЕС. Програмні нотифікації подають сигнали про стан роботи ПО, можливі помилки, чи навпаки успішне виконання певний дій. Екран зображений на рис. 2.6.8.

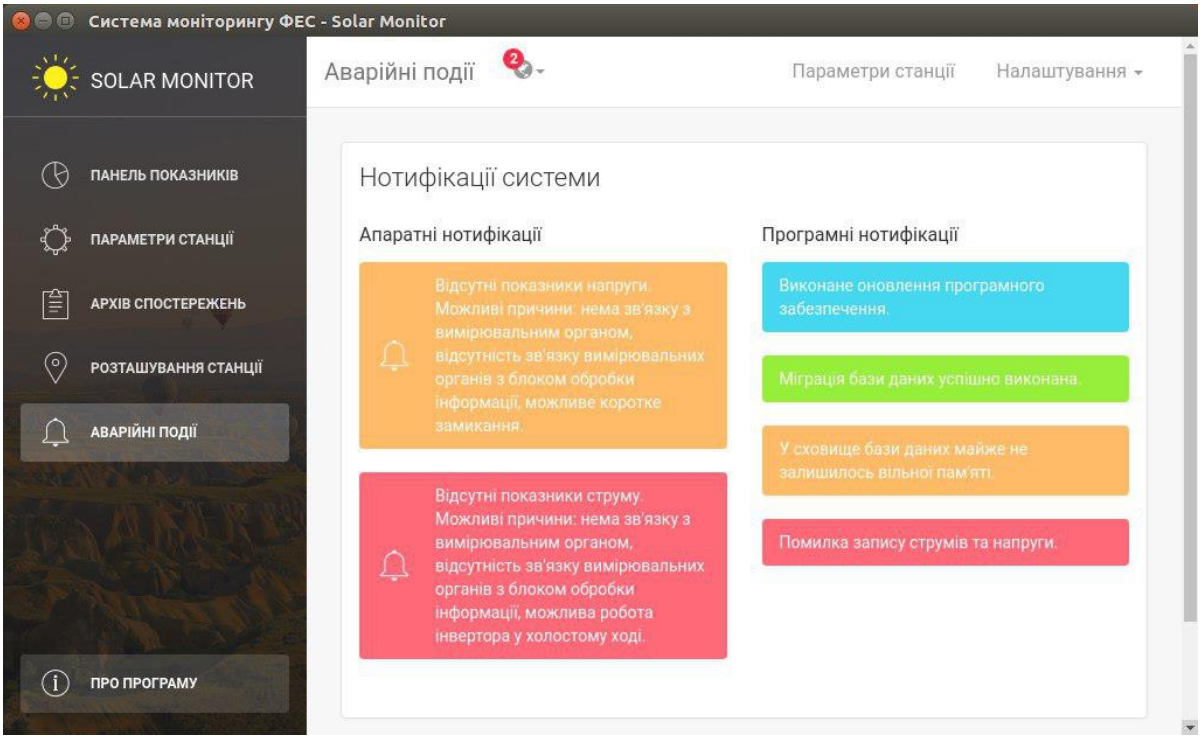


Рисунок 2.6.7 – Четвертий екран розробленої програми – екран нотифікацій

2.7. Висновки за розділом

Отже, в ході роботи над другим розділом диплому було виконано наступне:

- визначені певні технічні вимоги щодо розробки власної автоматизованої системи моніторингу роботи ФЕС;
- виконаний загальний огляд компонентів запропонованої системи;
- описані алгоритми роботи кожного функціонального блоку системи та взаємодії між ними.

Результатом роботи над даним розділом стали

- розроблена фізична модель моніторингової системи;
- створено програмне забезпечення для моніторингової системи;

- створено апаратне забезпечення розробленої системи моніторингу електричних характеристик роботи ФЕС та метеорологічних параметрів.

Розробка являє собою цілу екосистему, адже покриває весь цикл роботи, починаючи від збору даних з фізично віддалених датчиків та приладів до програмної обробки інформації з подальшою можливістю додавання технічного аналізу та прогнозування.

Польові випробування розробленої системи проводились на базі київського офісу компанії «Атмосфера», що є дистриб'ютором обладнання ВДЕ. Розроблена система монтувалась на діюче обладнання ФЕС, що дало змогу зняти параметри діючої напруги, струму, активної та реактивної потужностей. Таким чином було здійснено перевірку роботи як апаратної, так і програмної частини розробленої системи моніторингу роботи ФЕС.